IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No: 10/662,889	§		
Filed: September 15, 2003	§	Examiner:	Wang, Liang Che A.
Inventor(s):	§	Group/Art Unit:	2155
Syed Mohammed Amir Husain, Todd John Enright, Barry W. Thornton	§	Atty. Dkt. No:	5602-11500
Title: SYSTEM AND METHOD FOR REDUCING USER-APPLICATION INTERACTIONS TO ARCHIVABLE FORM	§		

APPEAL BRIEF

Dear Sir or Madam:

Further to the Notice of Appeal filed on January 14, 2008, Appellants present this Appeal Brief. Appellants respectfully request that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by ClearCube Technology, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 8834 Capitol of Texas Highway, Austin, TX 78759, as evidenced by the assignment recorded at Reel 014869, Frame 0733.

II. RELATED APPEALS AND INTERFERENCES

Related cases having application serial numbers 10/662,936 and 10/662,955 are also under appeal. There are no other related appeals or interferences known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1, 2, 4-13, 15-24, and 26-33 are pending. Claims 3, 14, and 25 are canceled. Claims 1, 2, 4-13, 15-24, and 26-33 are rejected, and the rejection of these claims is being appealed. A copy of claims 1, 2, 4-13, 15-24, and 26-33 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed towards a method comprising entering user input to a source application on a first computer system (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 22, reference characters 101A, 101B) to request performance of a task (*see*, e.g., page 6, lines 6-9; page 55, lines 12-17). The method also comprises performing the task on the first computer system in response to the user input (*see*, e.g., page 6, line 9). The method further comprises generating a message in response to the user input (*see*, e.g., Fig. 23, reference character 2401; page 6, lines 6-9; page 55, lines 12-17). The message comprises one or more instructions which are computer-executable to perform the task (*see*, e.g., page 6, lines 10-11). The message comprises metadata which comprise identifying characteristics of the source application (*see*, e.g., page 6, lines 11-12; page 55, lines 21-23). Additionally, the method comprises translating the message from an original format to a portable format on the first computer system, thereby generating a portable message (*see*, e.g., Fig. 23, reference character 2403; page 6, lines 12-15; page 55, lines 18-23). The method also comprises storing the portable message in a message log (*see*, e.g., Fig. 23, reference character 2405; page 6, lines 16-17; page 55, lines 24-28). The method further comprises retrieving the portable message from the message log (*see*, e.g., page 6, lines 16-18; page 55, lines 27-28). The method also comprises executing the one or more instructions to perform the task (*see*, e.g., Fig. 23, reference character 2411; page 6, lines 16-18; page 55, lines 27-28; page 56, lines 14-21; page 60, lines 11-13) on one or more additional computer systems (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 22, reference characters 101A, 101B).

Independent claim 12 is directed towards a computer-readable storage medium comprising program instructions (*see*, e.g., page 63, lines 1-6). The program instructions are computer-executable to implement receiving user input at a source application on a first computer system (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 22, reference characters 101A, 101B) to request performance of a task(*see*, e.g., page 6, lines 6-9; page 55, lines 12-17). The program instructions are also computer-executable to implement performing the task on the first computer system in response to the user input

(*see*, e.g., page 6, line 9). The program instructions are further computer-executable to implement generating a message in response to the user input (*see*, e.g., Fig. 23, reference character 2401; page 6, lines 6-9; page 55, lines 12-17). The message comprises one or more instructions which are computer-executable to perform the task (*see*, e.g., page 6, lines 10-11). The message comprises metadata which comprise identifying characteristics of the source application (*see*, e.g., page 6, lines 11-12; page 55, lines 21-23). Additionally, the program instructions are computer-executable to implement translating the message from an original format to a portable format on the first computer system, thereby generating a portable message (*see*, e.g., Fig. 23, reference character 2403; page 6, lines 12-15; page 55, lines 18-23). The program instructions are also computer-executable to implement storing the portable message in a message log (*see*, e.g., Fig. 23, reference character 2405; page 6, lines 16-17; page 55, lines 24-28). The program instructions are further computer-executable to implement retrieving the portable message from the message log (*see*, e.g., page 6, lines 16-18; page 55, lines 27-28). The program instructions are also computer-executable to implement executing the one or more instructions to perform the task (*see*, e.g., Fig. 23, reference character 2411; page 6, lines 16-18; page 55, lines 27-28; page 56, lines 14-21; page 60, lines 11-13) on one or more additional computer systems (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 22, reference characters 101A, 101B).

Independent claim 23 is directed towards a system comprising a CPU and a memory which is coupled to the CPU (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 3, reference characters 304, 206; Fig. 22, reference characters 101A, 101B; page 15, line 21 to page 20, line 14). The memory stores program instructions (*see*, e.g., page 63, lines 1-6) which are executable by the CPU to receive user input at a source application to request performance of a task.(*see*, e.g., page 6, lines 6-9; page 55, lines 12-17). The program instructions are also executable by the CPU to perform the task in response to the user input (*see*, e.g., page 6, line 9). The program instructions are further executable by the CPU to generate a message in response to the user input (*see*, e.g., Fig. 23, reference character 2401; page 6, lines 6-9; page 55, lines 12-17). The message comprises one or more instructions which are computer-executable to perform the task

(*see*, e.g., page 6, lines 10-11). The message comprises metadata which comprise identifying characteristics of the source application (*see*, e.g., page 6, lines 11-12; page 55, lines 21-23). Additionally, the program instructions are also executable by the CPU to translate the message from an original format to a portable format on the first computer system, thereby generating a portable message (*see*, e.g., Fig. 23, reference character 2403; page 6, lines 12-15; page 55, lines 18-23). The program instructions are also executable by the CPU to store the portable message in a message log (*see*, e.g., Fig. 23, reference character 2405; page 6, lines 16-17; page 55, lines 24-28). The system further comprises a second CPU and a second memory which is coupled to the second CPU (*see*, e.g., Fig. 1, reference characters 101, 105, 109; Fig. 3, reference characters 304, 206; Fig. 22, reference characters 101A, 101B; page 15, line 21 to page 20, line 14). The second memory stores additional program instructions (*see*, e.g., page 63, lines 1-6) which are executable by the second CPU to execute the one or more instructions in the portable message to perform the task again (*see*, e.g., Fig. 23, reference character 2411; page 6, lines 16-18; page 55, lines 27-28; page 56, lines 14-21; page 60, lines 11-13).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 4, 9-12, 15, 20-23, 26, and 31-33 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Kenton (U.S. Patent No. 6,845,507).

2. Claims 2, 13, and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kenton in view of Dodrill, et al. (U.S. Patent No. 6,766,298, hereinafter "Dodrill").

3. Claims 5-8, 16-19, and 27-30 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kenton in view of Sweeney, et al. (U.S. Patent Application Publication No. 2002/0032646, hereinafter "Sweeney").

VII. ARGUMENT

First Ground of Rejection:

Claims 1, 4, 9-12, 15, 20-23, 26, and 31-33 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Kenton (U.S. Patent No. 6,845,507). Appellants traverse this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 4, 9, 11, 12, 15, 20, 22, 23, 26, 31, and 33:

Appellants respectfully submit that Kenton does not teach or suggest a method comprising "performing the task on the first computer system in response to the user input" and "retrieving the portable message from the message log; and executing the one or more instructions to perform the task on one or more additional computer systems" in combination with the remaining features of claim 1. Kenton discloses a system for workflow automation in which workflow instructions are accessed from messages passed from one application to another (see, e.g., col. 3, line 47 to col. 4, line 2). Figs. 2 and 3 illustrate examples of workflows including various workflow tasks. Kenton also discloses that different tasks in a workflow may be performed respectively on different computer systems (see, e.g., Fig. 2). However, Kenton does not teach or suggest that the same task is performed on at least two different computer systems.

In the "Response to Arguments" section of the Final Office Action, the Examiner contends that Kenton performs the same task regarding the allocation of stock shares on at least two different computer systems. Appellants respectfully disagree with this characterization of Kenton's teachings. On a first computer system running a Trade Management Application (TMA), a financial transaction is performed to allocate shares of stock to various client portfolios (see, e.g., col. 4, lines 35-39 and 45-47). On a second computer system running a Portfolio Management Application (PMA), a database transaction is perform to update a PMA database to reflect the stock allocation (see, e.g., col. 4, lines 39-41 and 61-64). However, the financial transaction and the database transaction are two different tasks. Therefore, Kenton does not teach or suggest a method

comprising "executing the one or more instructions to perform the task on one or more additional computer systems" (emphasis added) in combination with the remaining features of claim 1.

Furthermore, Appellants respectfully submit that Kenton does not teach or suggest a method comprising "wherein the message comprises metadata which comprise identifying characteristics of the source application" in combination with the remaining features of claim 1. The Final Office Action argues that this limitation is taught by Kenton in col. 7, lines 48-58. Appellants respectfully disagree. In the cited passage, Kenton discloses that a workflow message contains hierarchical data including a <Stage> tag that encloses a list of data needed for a task. Kenton further describes the tagged data as "data needed by the steps that a particular application must execute to fulfill its role in the process." However, there is no teaching or suggestion in Kenton that the message identifies the particular application or otherwise comprises identifying characteristics of the source application. Although it is argued in the "Response to Arguments" section of the Final Office Action that the TMA "is part of the stage elements," Appellants can find no teaching or suggestion that the <Stage> tag or any other metadata in Kenton actually identifies the TMA.

Anticipation requires the presence of each and every limitation of the claimed invention, arranged as in the claim, in a single prior art reference. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Kenton fails to disclose a method comprising "performing the task on the first computer system in response to the user input," "retrieving the portable message from the message log," and "executing the one or more instructions to perform the task on one or more additional computer systems," "wherein the message comprises metadata which comprise identifying characteristics of the source application," in combination with the remaining features of claim 1. Therefore, Kenton cannot be said to anticipate claim 1.

Accordingly, claim 1 and its dependent claims 1, 4, 9, and 11 are believed to patentably distinguish over Kenton for at least the reasons given above. Claims 12 and 23 recite features similar to those of claim 1 and, along with their dependent claims 15, 20, 22, 26, 31, and 33, are believed to patentably distinguish over Kenton for at least the same reasons.

Claims 10, 21, and 32:

Regarding claim 10, Appellants respectfully submit that Kenton does not teach or suggest a method comprising "routing the portable message to a target application on the second computer system based on the metadata" in combination with the features of the base claim 1 and intervening claim 9. The Final Office Action argues that these limitations are taught by Kenton in col. 7, lines 55-58 and Figures 2 and 4. Appellants respectfully disagree. In the cited passage, Kenton discloses that a workflow message contains hierarchical data including a <Stage> tag that encloses a list of data needed for a task. Kenton further describes the tagged data as "data needed by the steps that a particular application must execute to fulfill its role in the process." However, there is no teaching or suggestion in Kenton that the message is routed to a target application on the second computer system based on the metadata.

Anticipation requires the presence of each and every limitation of the claimed invention, arranged as in the claim, in a single prior art reference. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Kenton fails to disclose a method comprising "routing the portable message to a target application on the second computer system based on the metadata" in combination with the features of the base claim 1 and intervening claim 9. Therefore, Kenton cannot be said to anticipate claim 10.

Accordingly, claim 10 is believed to patentably distinguish over Kenton for at least the reasons given above. Claims 21 and 32 recite features similar to those of claim 10 and are believed to patentably distinguish over Kenton for at least the same reasons.

Second Ground of Rejection:

Claims 2, 13, and 24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kenton in view of Dodrill, et al. (U.S. Patent No. 6,766,298, hereinafter "Dodrill"). Appellants traverse this rejection for the following reasons.

Claims 2, 13, and 24:

Regarding claim 2, Appellants respectfully submit that the cited references, taken individually or in combination, do not teach or suggest a method comprising "retrieving the portable message from the message log" and "executing the one or more instructions to perform the task again on the first computer system" in combination with the features of the base claim 1. The Final Office Action argues that these limitations are taught by Dodrill in col. 15, lines 16-18. Appellants respectfully disagree. In the cited passage, Dodrill discloses that the same XML document may be executed more than once. However, there is no teaching or suggestion in Dodrill that the XML document is retrieved from a message log.

To establish a *prima facie* case of obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. For at least the reasons discussed above with respect to claim 1, Appellants respectfully submit that Kenton and Dodrill, taken individually or in combination, would not produce all the limitations recited in claim 2 and its base claim 1.

Furthermore, Appellants respectfully submit that it would not have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kenton in

the manner suggested by the Examiner. In particular, Appellants respectfully submit that there is no suggestion or reason to combine the teachings of Kenton and Dodrill. Kenton discloses a multi-stage workflow using a message queue, while Dodrill discloses a web-based voice messaging system in which XML documents are processed in audio operations. The Final Office Action argues that the combination would have been obvious "because having the XML to be executed and prompt to the user again would allow user to correct a previous error as taught by Dodrill." However, Kenton's tasks involve financial transactions and database operations, not the obtaining of user input via voice commands as taught by Dodrill. Thus, Appellants can see no reason to combine the voice-based input techniques of Dodrill with the system disclosed by Kenton in the manner suggested by the Examiner.

Accordingly, Appellants respectfully submit that claim 2 is patentably distinct from the cited references. For similar reasons, claims 13 and 24 are believed to patentably distinguish over the cited references.

Third Ground of Rejection:

Claims 5-8, 16-19, and 27-30 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kenton in view of Sweeney, et al. (U.S. Patent Application Publication No. 2002/0032646, hereinafter "Sweeney"). Appellants traverse this rejection for the following reasons.

Claims 5-8, 16-19, and 27-30:

Regarding claim 5, Appellants respectfully submit that the cited references, taken individually or in combination, do not teach or suggest a method comprising "sorting the message log by one or more elements of the metadata" in combination with the features of the base claim 1. The Final Office Action argues that these limitations are taught by Sweeney in paragraph [0069]. Appellants respectfully disagree. In the cited passage, Sweeney discloses that communications (e.g., e-mail and "a-mail") in an in-box may be

sorted or filtered by date, sender/recipient, or communication type. However, there is no teaching or suggestion in Sweeney that a communication in the in-box is a message stored in a message log as recited in the base claim 1. Additionally, there is no teaching or suggestion in Sweeney that the communications are sorted by elements of the metadata which comprise identifying characteristics of the source application as recited in the base claim 1.

To establish a *prima facie* case of obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. For at least the reasons discussed above with respect to claim 1, Appellants respectfully submit that Kenton and Sweeney, taken individually or in combination, would not produce all the limitations recited in claim 5 and its base claim 1.

Furthermore, Appellants respectfully submit that it would not have been obvious to one of ordinary skill in the art at the time the invention was made to modify Kenton in the manner suggested by the Examiner. In particular, Appellants respectfully submit that there is no suggestion or reason to combine the teachings of Kenton and Sweeney. Kenton discloses a multi-stage workflow using a message queue. A queue is a type of data structure in which items are stored in a particular order (i.e., first-in, first-out [FIFO]). Applicant can find no evidence in Kenton that it would be advantageous to change the FIFO order of items in the queue (e.g., by sorting the items in the queue). Thus, Appellants can see no reason to combine the sorting techniques of Sweeney with the message queue disclosed by Kenton in the manner suggested by the Examiner.

Accordingly, Appellants respectfully submit that claim 5 is patentably distinct from the cited references. For similar reasons, claims 6, 7, 8, 16-19, and 27-30 are believed to patentably distinguish over the cited references.

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1, 2, 4-13, 15-24, and 26-33 was erroneous, and reversal of the decision is respectfully requested.

The fee of $255.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Appellant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5602-11500/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: March 13, 2008 JCH/RPH

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method comprising:

entering user input to a source application on a first computer system to request performance of a task;

performing the task on the first computer system in response to the user input;

generating a message in response to the user input, wherein the message comprises one or more instructions which are computer-executable to perform the task, and wherein the message comprises metadata which comprise identifying characteristics of the source application;

translating the message from an original format to a portable format on the first computer system, thereby generating a portable message;

storing the portable message in a message log;

retrieving the portable message from the message log; and

executing the one or more instructions to perform the task on one or more additional computer systems.

2. The method of claim 1, further comprising:

retrieving the portable message from the message log; and

executing the one or more instructions to perform the task again on the first computer system.

4. The method of claim 1,

wherein the message log comprises a queue.

5. The method of claim 1, further comprising:

sorting the message log by one or more elements of the metadata.

6. The method of claim 1, further comprising:

sorting the message log by application type of the source application.

7. The method of claim 1, further comprising:

sorting the message log by message sender.

8. The method of claim 1, further comprising:

sorting the message log by message recipient.

9. The method of claim 1, further comprising:

sending the portable message from the first computer system to a second computer system using peer-to-peer message passing between the first computer system, the second computer system, and optionally one or more intermediary computer systems; and

performing the requested task on the second computer system.

10. The method of claim 9, further comprising:

routing the portable message to a target application on the second computer system based on the metadata.

11. The method of claim 1,

wherein the portable format comprises XML, and wherein the portable message comprises an XML message.

12. A computer-readable storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

receiving user input at a source application on a first computer system to request performance of a task;

performing the task on the first computer system in response to the user input;

generating a message in response to the user input, wherein the message comprises one or more instructions which are computer-executable to perform the task,

and wherein the message comprises metadata which comprise identifying characteristics of the source application;

translating the message from an original format to a portable format on the first computer system, thereby generating a portable message;

storing the portable message in a message log;

retrieving the portable message from the message log; and

executing the one or more instructions to perform the task on one or more additional computer systems.

13. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

retrieving the portable message from the message log; and

executing the one or more instructions to perform the task again on the first computer system.

15. The computer-readable storage medium of claim 12,

wherein the message log comprises a queue.

16. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

sorting the message log by one or more elements of the metadata.

17. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

sorting the message log by application type of the source application.

18. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

sorting the message log by message sender.

19. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

sorting the message log by message recipient.

20. The computer-readable storage medium of claim 12, wherein the program instructions are further computer-executable to implement:

sending the portable message from the first computer system to a second computer system using peer-to-peer message passing between the first computer system, the second computer system, and optionally one or more intermediary computer systems; and

performing the requested task on the second computer system.

21. The computer-readable storage medium of claim 20, wherein the program instructions are further computer-executable to implement:

routing the portable message to a target application on the second computer system based on the metadata.

22. The computer-readable storage medium of claim 12,

wherein the portable format comprises XML, and wherein the portable message comprises an XML message.

23. A system comprising:

a CPU;

a memory which is coupled to the CPU, wherein the memory stores program instructions which are executable by the CPU to:

receive user input at a source application to request performance of a task;

perform the task in response to the user input;

generate a message in response to the user input, wherein the message comprises one or more instructions which are computer-executable to perform the task, and wherein the message comprises metadata which comprise identifying characteristics of the source application;

translate the message from an original format to a portable format on the first computer system, thereby generating a portable message; and

store the portable message in a message log;

a second CPU; and

a second memory which is coupled to the second CPU, wherein the second memory stores additional program instructions which are executable by the second CPU to execute the one or more instructions in the portable message to perform the task again.

24. The system of claim 23, wherein the program instructions are further executable by the CPU to:

retrieve the portable message from the message log; and

execute the one or more instructions to perform the task again.

26. The system of claim 23,

wherein the message log comprises a queue.

27. The system of claim 23, wherein the program instructions are further executable by the CPU to:

sort the message log by one or more elements of the metadata.

28. The system of claim 23, wherein the program instructions are further executable by the CPU to:

sort the message log by application type of the source application.

29. The system of claim 23, wherein the program instructions are further executable by the CPU to:

sort the message log by message sender.

30. The system of claim 23, wherein the program instructions are further executable by the CPU to:

sort the message log by message recipient.

31. The system of claim 23, wherein the program instructions are further executable to:

send the portable message to a second computer system using peer-to-peer message passing between a first computer system, a second computer system, and optionally one or more intermediary computer systems; and

perform the requested task on the second computer system.

32. The system of claim 31, wherein the program instructions are further executable to:

route the portable message to a target application on the second computer system based on the metadata.

33. The system of claim 23,

wherein the portable format comprises XML, and wherein the portable message comprises an XML message.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131, or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no decisions rendered by a court or the Board in any related proceedings known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.